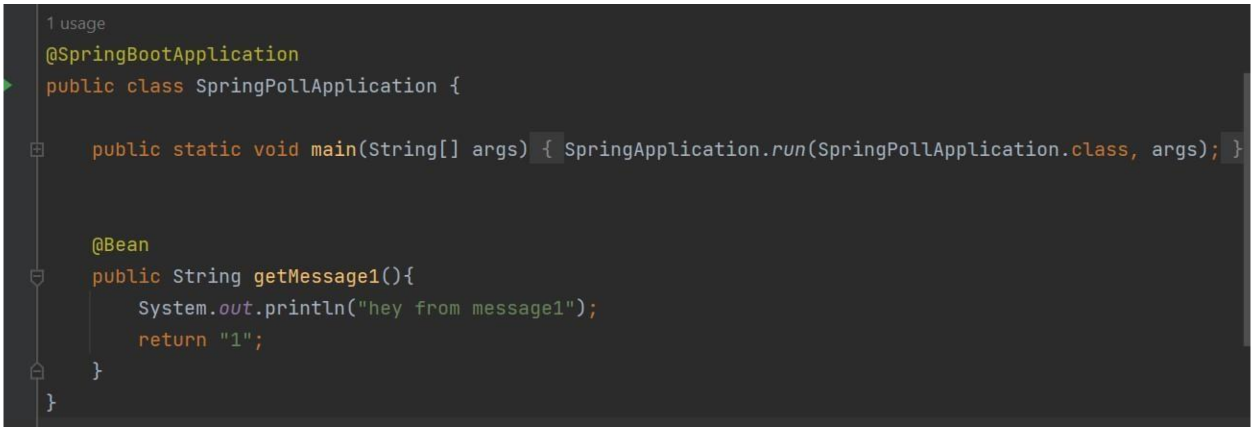
Exercise (Spring Container)

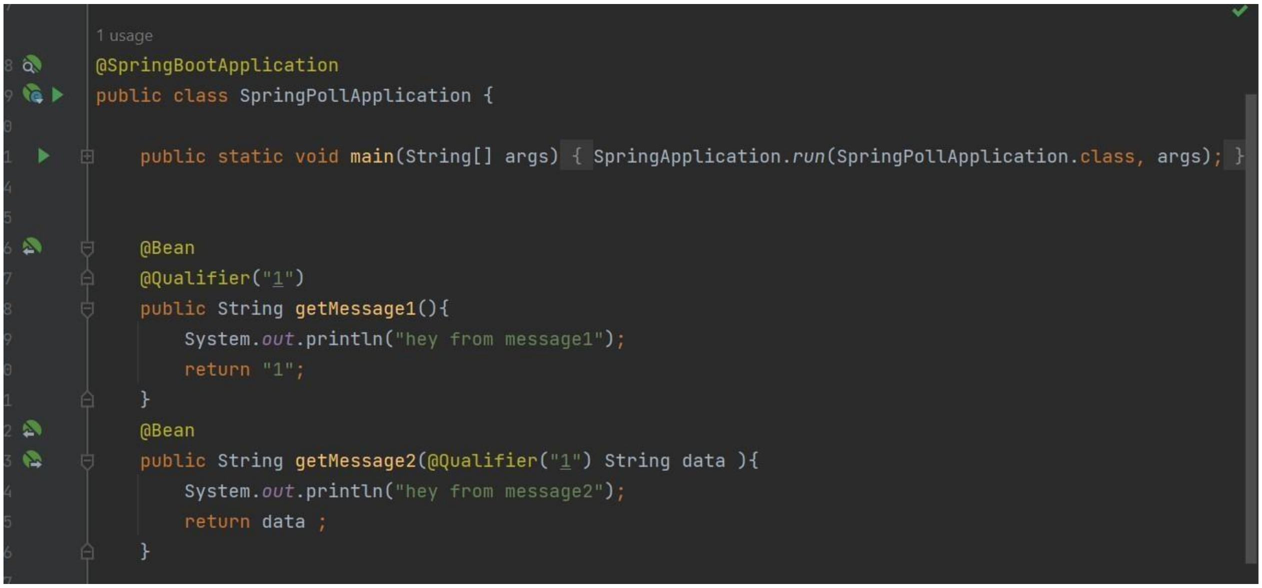
What is the output ? Q1



```
1 usage
@SpringBootApplication
public class SpringPollApplication {

    public static void main(String[] args) { SpringApplication.run(SpringPollApplication.class, args); }


    @Bean
    public String getMessage1(){
        System.out.println("hey from message1");
        return "1";
    }
}
```

- Output:
  - This will print "hey from message1". Then it will place the String "1" into the Spring container.

Q2

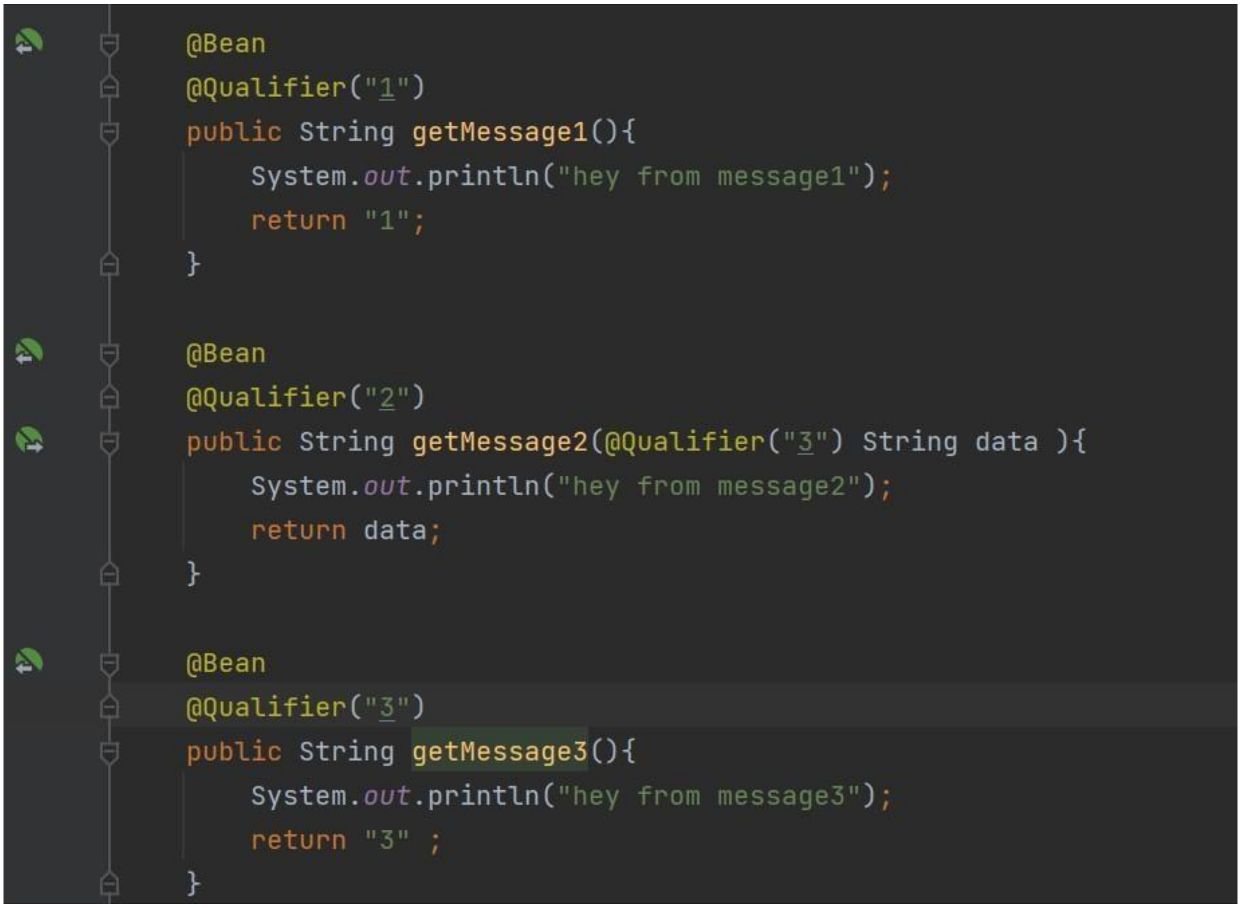


```
1 usage
@SpringBootApplication
public class SpringPollApplication {

    public static void main(String[] args) { SpringApplication.run(SpringPollApplication.class, args); }


    @Bean
    @Qualifier("1")
    public String getMessage1(){
        System.out.println("hey from message1");
        return "1";
    }
    @Bean
    public String getMessage2(@Qualifier("1") String data ){
        System.out.println("hey from message2");
        return data ;
    }
```

- Output:
    - This will print "hey from getMessage1" and place the String "1" into the Spring container.

    - getMessage2 will call getMessage1, receive "1" as its return value, print "hey from message2", and return "1" back into the container.
      //getMessage2 depends on getMessage1, so it first executes getMessage1 to get its result, then processes it and stores it in the container.

Q3



```
@Bean
@Qualifier("1")
public String getMessage1(){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```

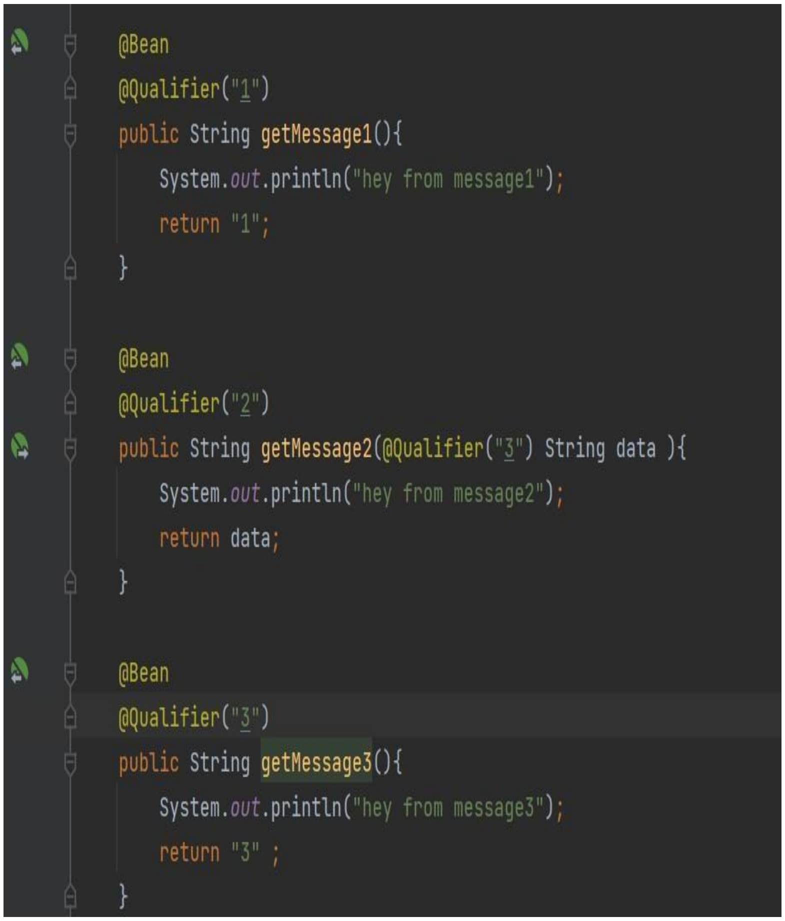
Output:

- This will print "hey from getMessage1" and place the String "1" into the Spring container.

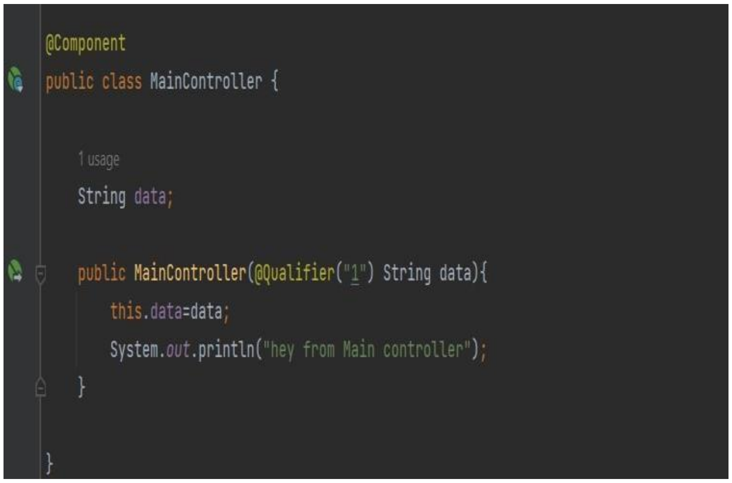
- getMessage3 will then execute, print "hey from message3", and return "3" to the container.
  //this will run next because getMessage2 have dependencies on getMessage3

- getMessage2 will call getMessage3, store "3" in the variable data, print "hey from message2", and place data ("3") into the container.
  //getMessage2 depends on getMessage3, so it first executes getMessage3 to get its result, then processes it and stores it in the container.

Q4



```
@Bean
@Qualifier("1")
public String getMessage1(){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```

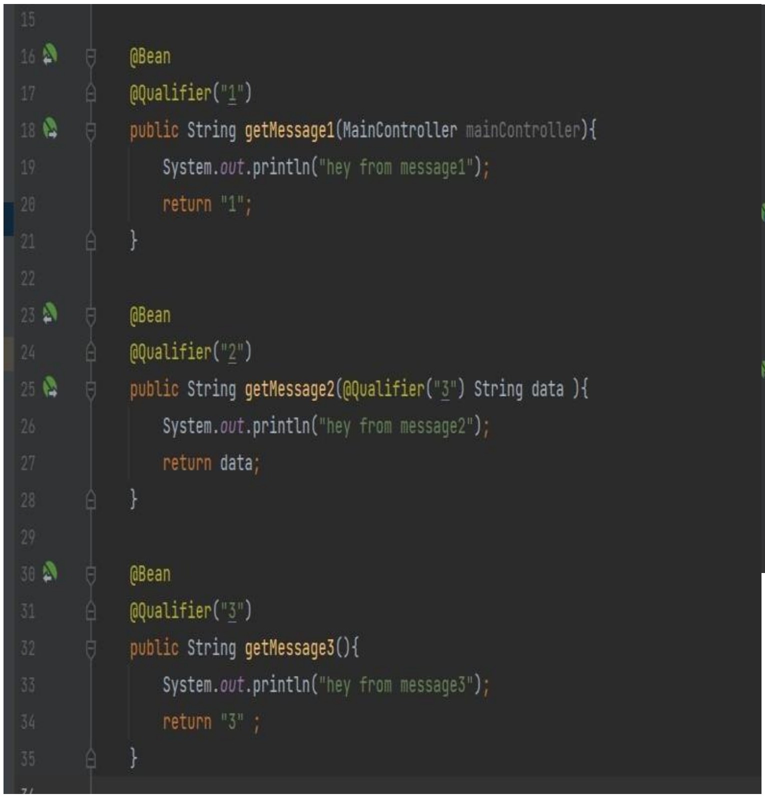


```
@Component
public class MainController {

    String data;

    public MainController(@Qualifier("1") String data){
        this.data=data;
        System.out.println("hey from Main controller");
    }

}
```

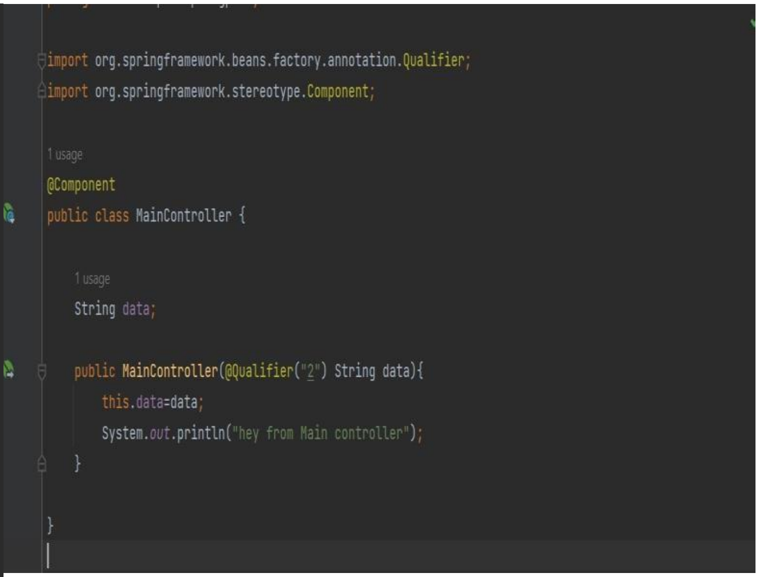
Output:

- This will print "hey from getMessage1" and place the String "1" into the Spring container.
- The MainController will call getMessage1, store "1" in the variable data, and print "hey from Main controller".
- getMessage3 will execute, print "hey from message3", and place "3" into the container.
- getMessage2 will call getMessage3, store "3" in the variable data, print "hey from message2", and return data ("3") to the container.

Q5



```java
@Bean
@Qualifier("1")
public String getMessage1(MainController mainController){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```



```java
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;

1 usage
@Component
public class MainController {

    1 usage
    String data;

    public MainController(@Qualifier("2") String data){
        this.data=data;
        System.out.println("hey from Main controller");
    }

}
```

Output:

- getMessage3 will execute, print "hey from message3", and return "3" to the container since it has no dependencies.

- getMessage2 will call getMessage3, store "3" in the variable data, print "hey from message2", and place data ("3") into the container.

- The MainController will take getMessage2, data ("3" from getMessage3), and print "hey from Main controller".
  // MainController depends on getMessage2, so it first executes getMessage3 to get its result, after that executes getMessage2.

- getMessage1 will then print "hey from getMessage1", create a MainController object, and place "1" into the container.